Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)

Submission Title: [FEC design details for merged baseline draft]
Date Submitted: [17 December, 2009]
Source: [Sridhar Rajagopal, Shadi Abu Surra] [Samsung Electronics]

Abstract: Provides details for RS code design for merged baseline draft

Purpose: [Contribution to IEEE 802.15.7 VLC TG]

Data rates – PHY 1

	Optical rate	Modulation	Line coding	FEC	Data rate
PHY I	200 kHz	OOK	Manchester	1/16	6.25 kbps
	200 kHz	OOK	Manchester	1/8	12.5 kbps
	200 kHz	OOK	Manchester	1/4	25 kbps
	200 kHz	OOK	Manchester	1/2	50 kbps
	200 kHz	OOK	Manchester	1	100 kbps
	600 kHz	VPM	4B6B	1/2	200 kbps
	600 kHz	VPM	4B6B	1	400 kbps

Data rates – PHY II

	Optical rate	Modulation	Line coding	FEC	Data rate
PHY II	6 MHz	VPM	4B6B	4/5	3.2 Mbps
	12 MHz	VPM	4B6B	4/5	6.4 Mbps
	24 MHz	OOK	8B10B	1/2	9.6 Mbps
	60 MHz	OOK	8B10B	1/2	24 Mbps
	60 MHz	OOK	8B10B	4/5	38.4 Mbps
	120 MHz	OOK	8B10B	4/5	76.8 Mbps
	120 MHz	OOK	8B10B	1	96 Mbps

Current agreed baseline draft details for FEC code

High rate PHY

- 4B6B
- 8B10B
- RS coding (shortened code support)

Low rate PHY

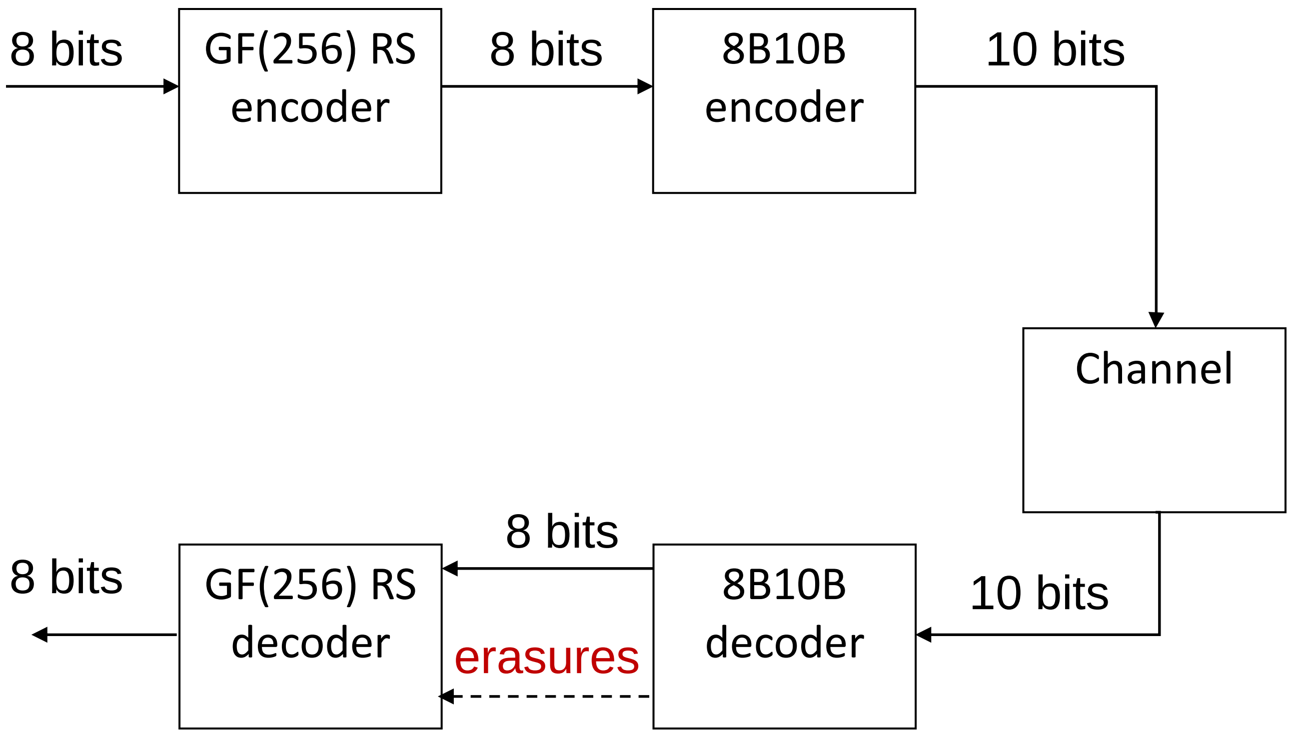
- Manchester
- 4B6B
- Concatenated RS (with Viterbi) under consideration

Presentation focus

Complete FEC design details <u>for high rate PHY</u>

- Mapping of 4B6B and 8B10B with the RS code
- Using a single RS encoder/decoder hardware
- **Performance evaluation**

Currently proposed (255, k) RS code



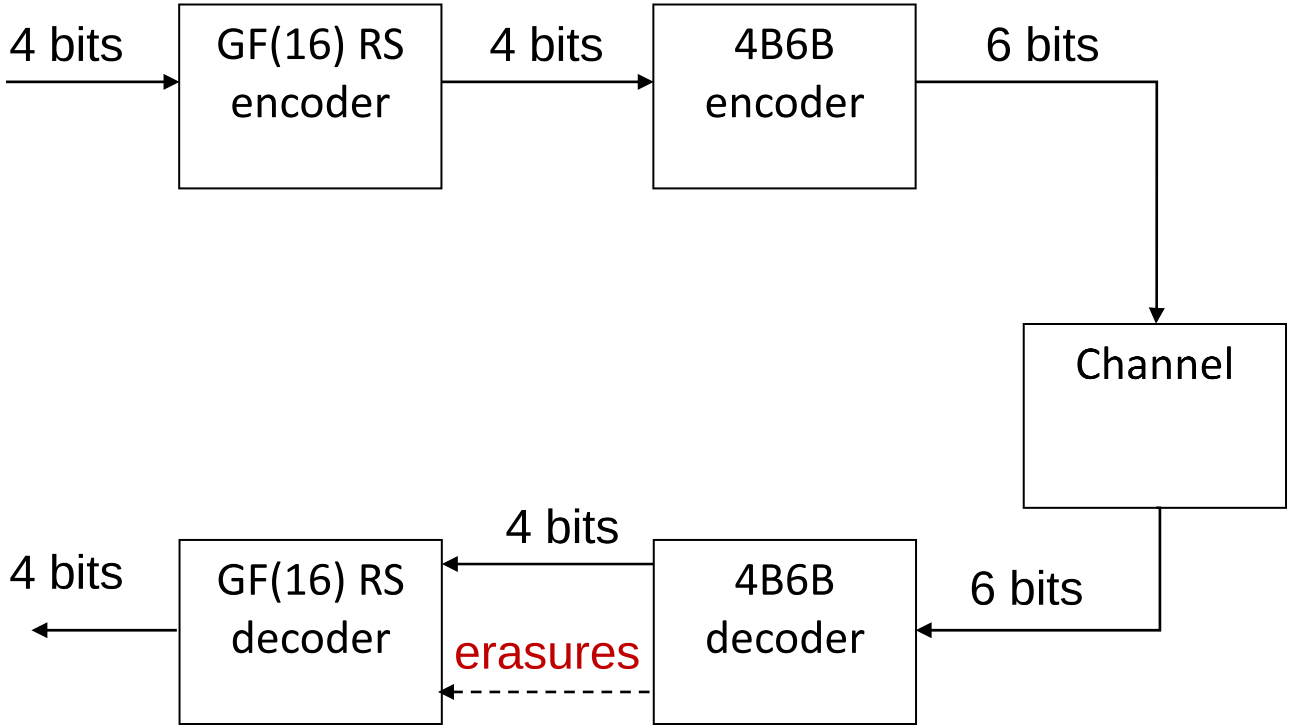
8b10b code naturally maps to the proposed (255,k) RS code since the code operates on GF(256), using 8 bits for a symbol

Design for 4B6B code

4B6B code requires 4 bit input

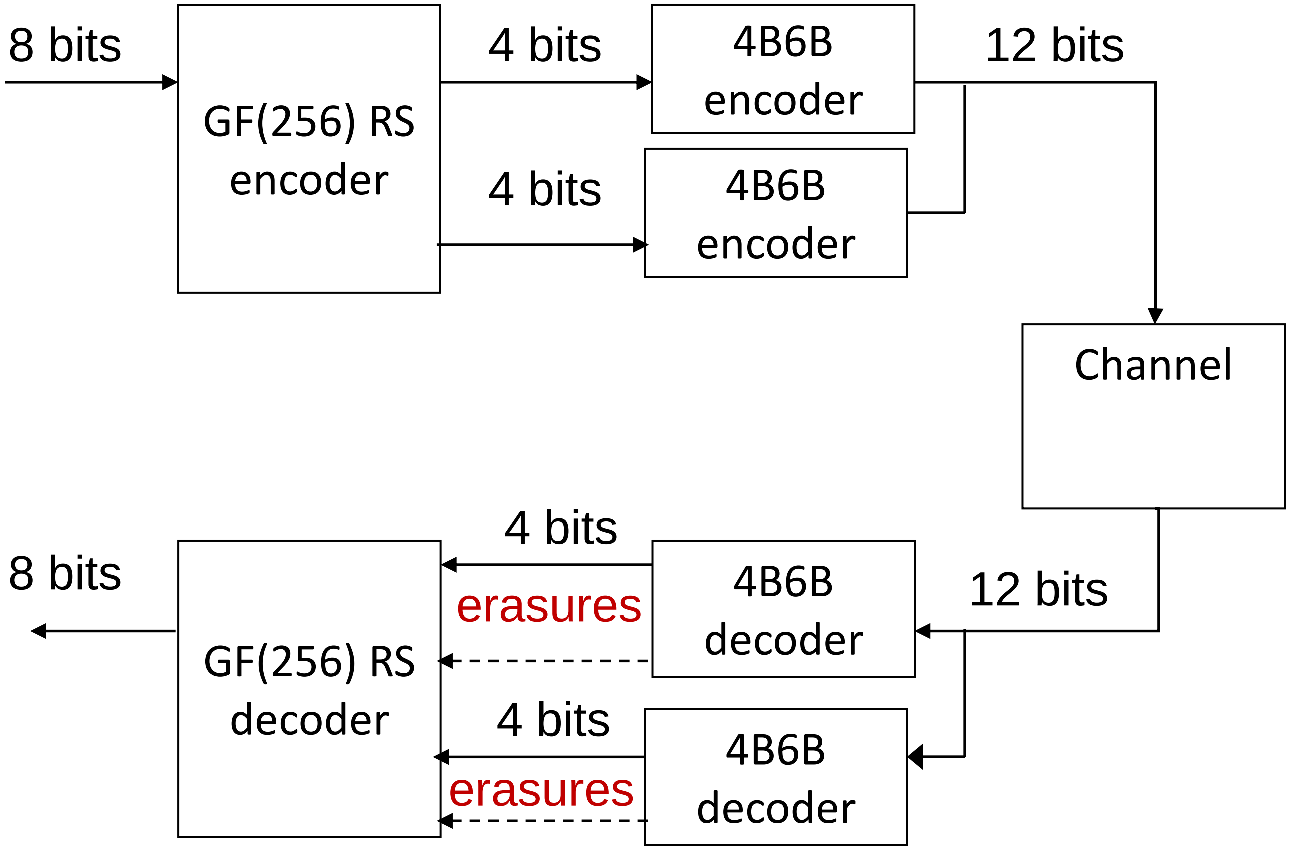
What is the best way to use 4B6B with a RS code?

Option 1: (15, k) RS code



This is the most straightforward option. However, it now requires a new [additional] encoder and decoder. Hence, we do not support this option.

Option 2: Use (255, k) RS code



Option 2

Looks very reasonable

However, does it provide optimal performance?

Issue:

- Even if one 4B6B line code is not decoded correctly (and the other 4B6B code is decoded correctly), entire RS symbol is in error
- We should be able to do better

Solution

Need to design a RS code such that it works as both GF(16) and GF(256) to provide optimal performance when 4B6B and 8B10B line codes are used respectively

Same hardware should be re-used for both cases

Is this possible?

Theorem

In the finite field GF(p^n), GF(p^m) is a subfield of GF(p^n) if and only if *m* divides *n*.

Note that 256 = 2^8 and 16 = 2^4. Since 4 divides 8, then GF(16) is a subfield of GF(256). This means that it is possible to use the GF(256) RS encoder and decoder hardware to process the GF(16) RS codes as well.

Mapping process

We first identify the elements in GF(256) which form GF(16). Then, we propose a mapping from the space F_2^4 (the space of 4-bits symbols) into these elements in GF(256) which form GF(16).

Consider GF(256) generated by $x^8+x^4+x^3+x^2+1$

let α be a primitive element in GF(256) (i.e., α generates all non-zero elements of GF(256) as $\alpha^0, \alpha^1, \ldots, \alpha^{254}$).

There are 15, and 255 non-zero elements in GF(16), and GF(256), respectively.

As 255/15 =17, then α^{17} generates all non-zero elements of GF(16) (i.e., $(\alpha^{17})^0, (\alpha^{17})^1, \ldots, (\alpha^{17})^{13}$ form the non-zero elements of GF(16)).

Mapping table

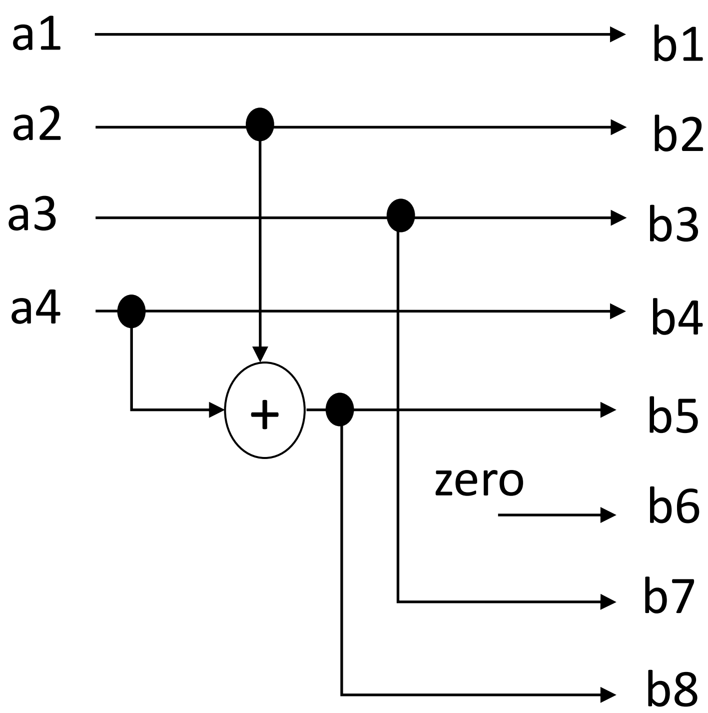
Power form (α^x)	additive form (α^x = b1 b2 … b8)							
	b1	b2	b3	b4	b5	b6	b7	b8
-1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
17	0	0	0	1	1	0	0	1
34	0	1	1	1	0	0	1	0
51	0	1	0	1	0	0	0	0
68	1	0	0	1	1	0	0	1
85	0	1	1	0	1	0	1	1
102	0	0	1	0	0	0	1	0
119	1	1	0	0	1	0	0	1
136	1	1	1	1	0	0	1	0
153	0	1	0	0	1	0	0	1
170	1	1	1	0	1	0	1	1
187	0	0	1	1	1	0	1	1
204	1	0	1	1	1	0	1	1
221	1	0	1	0	0	0	1	0
238	1	1	0	1	0	0	0	0

Note:

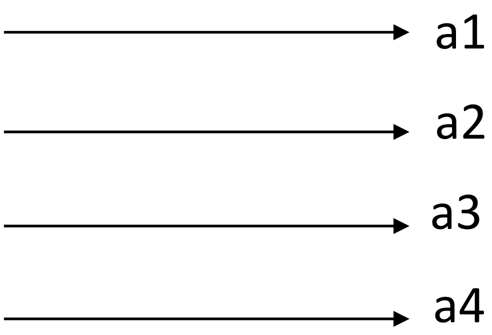
- b8 = b5
- b7 = b4
- b6 = 0
- b5 = b2+b4

Simple Mapping and Demapping

4 bits to 8 bits



8 bits to 4 bits

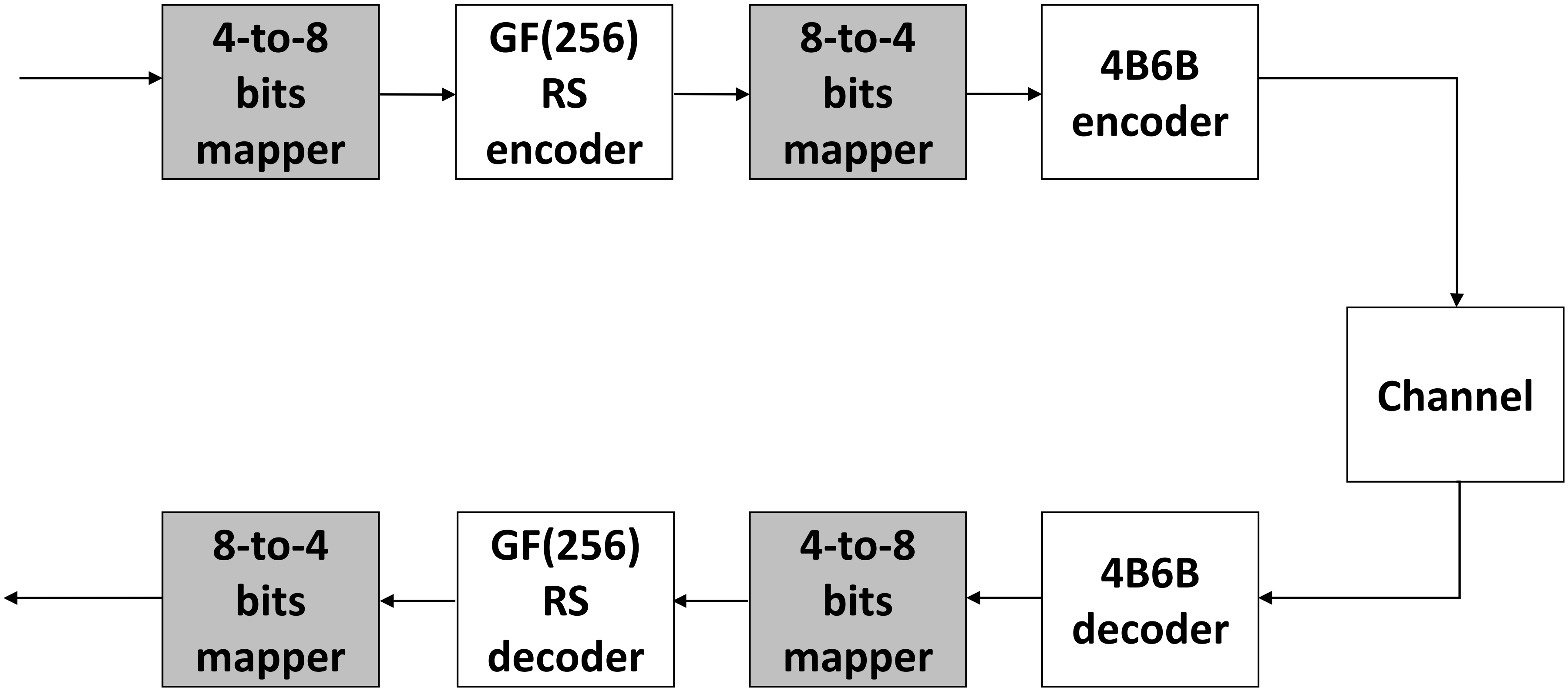


Notes

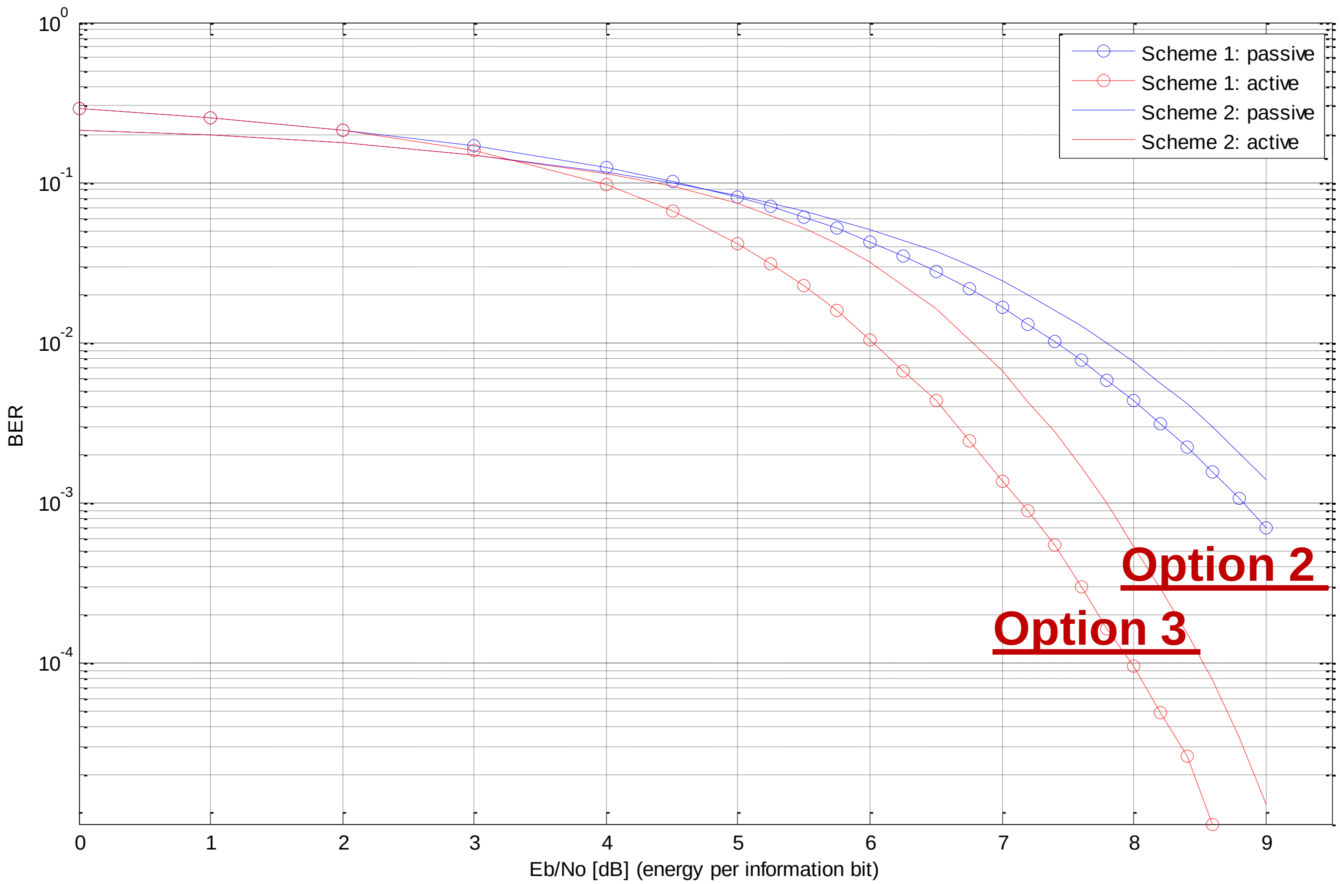
It is important to keep in mind that the set $\Psi = \{0, 1, \beta, \beta^2, \ldots, \beta^{13}\}$ are elements in GF(256) which form GF(16), where $\beta = \alpha^{17}$. And so, this set of elements is closed under addition, multiplication, and inverses.

Considering the proposed mapping, data symbols at the input of the Reed-Solomon encoder are in Ψ. Consequently, the codeword's symbols at the output of the Reed-Solomon encoder are in Ψ.

Option 3: GF(256) code operating as GF(16)



Performance comparisons for rate 4/5 code : RS (15,12) code



Benefits of Option 3

- ~ 0.6 dB performance gains
- Same RS encoder and decoder hardware for 4B6B and 8B10B code
- Decoder search can be reduced when searching over GF(16) in 4B6B mode of operation for low power
- Works with erasure information provided by 4B6B (or 8B10B) line code

Summary

We provide details on how 8B10B and 4B6B line codes should be used with the (255, k) RS code

- We support both line codes without any loss in performance
- We use the same hardware for both line coding schemes

Current baseline draft does not provide information on mapping line codes with FEC

We recommend using option 3 in merged baseline draft for mapping line codes with FEC

Recommendation to TG7 committee

Use FEC mapping for 4B6B line code for high rate PHY as defined on slides 17 and 15 for merged text draft

Further work is needed to look at code design for low rate PHY